PATIENT'S CHATTING MECHANISM

SERIOUS GAME FOR INTERNAL MEDICINE PRACTICE

LEE SIM SIEW

THESIS REPORT SUBMITTED IN FULFILMENT OF THE DEGREE OF COMPUTER SCIENCE

FACULTY OF COMPUTER SYSTEM AND SOFTWARE ENGINEERING

2013

ABSTRACT

This subject deals with an invention towards developing a Artificial Intelligent chatbot for a online serious game in the domain of medical and treatment. It lead to a creation of artificial intelligence chat-box which can communicate with user in human natural language. This robot is formulated to behave as a human patient. The artificial intelligent chatbot, Ill.Bot is being designed as a web-based language processor to analyze our human natural language by using fuzzy logic methods through web scripting. The trick of artificial intelligence is to locate the user input from the existing history or memory and match the response to related input through the use of search and match mechanism. Alongside the development of chatbot, an approaches, One-Match and All-Match Categories - algorithms and techniques had been used to enhance the process of keywords/pattern matching for chatbot.

ABSTRAK

Subjeck ini tertakluk dengan ciptaan ke arah pembangunan chatbot iaitu “Artificial Intelligent” untuk permainan talian serius dalam domain perubatan dan rawatan. Ia menuju kepada penciptaan kecerdasan buatan chat-kotak yang boleh berkomunikasi dengan pengguna dengan pengunaan bahasa manusia. Robot ini dirumus untuk bertindak sebagai pesakit manusia. Buatan pintar chatbot, Ill.Bot direka sebagai robot yang boleh manganalisi bahasa manusia berasaskan bahasa manusia dengan menggunakan kaedah logik kabur melalui web skrip Kecerdasan teknologi ini boleh bertindak balas kepada input dengan merujuk sejarah atau memori yang sedia ada dalam system Di samping itu, dengan pembangunan chatbot, One-Perlawanan dan Kategori-Match -penggunaan algoritma dan teknik digunakan untuk meningkatkan proses kata kunci / corak padanan untuk chatbot.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

1.1 Introduction

A serious game is a virtual environment designed for education, training or medical application - using innovative design mechanics and technology that engage the participant, promotes learning, and conveys knowledge. It takes advantage of game engines in order to improve the realistic experience of users. Serious games present in many areas of knowledge, including defense, manufacturing, education and medicine, among others. With advances in technologies, medical is now becomes the second only to the military in directing the evolution of serious games.

In virtual world, serious games can provide crisis resource training, with a large variety of cases, in a relatively cheap, readily available environment that provides a viable alternative to expensive simulators. It supplies training environments for disaster situations and mass casualty incidents, including combat care. Serious games allow multiple professionals to train simultaneously on one case and allow one professional to train multiple cases simultaneously. These skills are recognized as critical in reducing medical errors in dynamic high-risk environments, such as the operating room or emergency department. Games need to be designed to fit into residency teaching programs if they are to be used as a way of preventing medical errors. Simulation and serious gaming represent ideal teaching methods to optimize the knowledge and skill of residents before they are entrusted with procedures in real patients. Educators and games designers should develop serious games that train professionals in order to maximize patient safety.

Research into and developments of medical simulators have seen significant enhancements in recent times. A serious game is formally defined as an ‘interactive computer’ application. The use of Human–Computer Interaction (HCI) through such methods as affective computing to portray a mid-to-high level of fidelity is essential. Gaming and in the serious gaming perform most common type of interactions - dialogue system. Typically, this type interaction is done between human and interactive online characters, the use of virtual characters or so-called Non Player Characters through the support of Natural Language Processing (NLP). In order for game characters to be believable, NPCs must appear to possess qualities such as the ability to learn and adapt as well as being able to communicate in natural language.

Natural language processing (NLP) is defined as a field of computer science, artificial intelligence, and linguistic devoted to create computer system where interaction takes place between computer and the humans in the most natural way. It is an area of research and application that explores the ways where computers can be used to understand and manipulate natural language text or speech to perform actions. NLP researchers aim to collect knowledge on how human beings understand and use language, so to develop appropriate tools and techniques which will make computer systems interpret and manipulate natural languages to perform desired tasks. With available technologies, researchers aim to simulate real-world environment by creating interactive dialogue system.

The objective of our work is to design a conversational agent which capable of understanding natural language statements in a restricted semantic domain. This feature is intended to allow a natural dialogue, especially in the context of medical. This conversational agent will simulate a real-like-world communication and interaction for users to experiment real world environment in a medical situation. Therefore, a chatbot named ill.Bot is created to fulfill the requirement of this subject.

1.2 Problem Statement

Medical education faces a host of obstacles in coming decades requiring that it rethink the way it delivers medical curricula, especially with regard to critical thinking and differential diagnostics.

Traditional didactic curricula must be coupled with emerging technologies that provide experiential learning without risk to patients while not eroding the clinical effectiveness of advanced medical learners with the help of scenario practice in medical field.

Virtual-world technologies have to advance to a level where they must be considered as a method for delivering medical curricula effectively and safely; moreover, research must establish that such systems are reliable and valid means for delivering medical curricula.

1.3 Objective

The objectives of this research are:

- To come out with a creation of artificial intelligence chatbot which can communicate with user in human natural language.
- To use the algorithm of search and match method to analyze the user input and retrieve relevant responses.
- To make artificial intelligent robot posses the behaviour of a typical human being patient.

1.4 Scope

This research's scopes are:

- Web-based PHP application
- CSS styling interface design
- Using SQL database

2.0 LITERATURE REVIEW

Serious games engage players towards the acquisition of new skills rather than barely entertain them. Typically, serious games are designed for education, training or general counseling and assistance. But currently medical has became the second only to the military in directing the evolution of serious games (Illinios, 1976). This indicates that this application of digital games for training medical professional is on the rise. So-called 'serious' games for training tools that provide a challenging simulated environment, ideal for future surgical training (Dr M. P. Schijven, 2012). What could be more interesting than having communication with games like talking chatting with a real-life human beings? This incident could be happen with the participation of Natural Language Processing (Marilyn Walker, 2011). We present our current research activities associating natural language processing to serious games and virtual worlds by develop a reliable and valid learning platform delivering

medical curricula in virtual space for practitioners which tend to be simulation-based and used for training in medical field.

Natural Language Processing (NLP) enrich users' engagement to the game's world through interactive conversations (Cassell J. et al., 1999). Natural Language Processing (NLP) is an area of research and application that explores the ways where computers can be used to understand and manipulate natural language text or speech to perform actions. NLP researchers aim to collect knowledge on how human beings understand and use language, so to develop appropriate tools and techniques which will make computer systems interpret and manipulate natural languages to perform desired tasks (Gobinda G. Chowdhuryz, 2011). NLP have been proposed for representing conversational agent beliefs, desires and intentions. The levels of thrust, credibility and cooperativeness have been subject to a study in the last years, in which it can be collaborative or not (Roque, A., and Traum D.R., 2007). Our aim in the project is to provide natural language processing in serious games with conversation capabilities in collaborative environments via dialog where allow junior doctors to experience and train for a variety of acute medical scenarios using computerized mannequins as patients by the simulation of the real-world.

Dialogue systems are programs that can communicate with humans in natural language (Jurafsky, D. & Martin, J. H., 2001). A dialog is a verbal activity which involves at least two interlocutors and is used to accomplish a task in a given communication situation (Franck D., 2012). It is a coordinated sequence of actions (linguistic and non-linguistic) leading to a goal (Vernant, 1992). The system takes speech input and transforms it to a representation that can be understood by the system then the system will produce an appropriate response to the corresponding input (Jenny B., et al, 2007). Researchers believe that intelligent dialogue systems (also called advanced dialogue systems) may constitute a relevant answer to this problem (F. Dernoncourt, 2012). For example, if the business that we are interested in is a pharmacy or hospital, the dialogue between the simulated patient and the player, in this case a medical staff member, will help to get used to have dialogue with the patients and test their knowledge to solve usual, real-life situations that may be urgent and critical, where a mistake can be extremely serious, even fatal. In order to support conversations in serious games, it is necessity to implement a Dialogue Server (Jeremiah S., et al, 2011). Where it models goal-oriented conversations, in which

virtual characters know the type of knowledge the player must acquire to successfully complete a task in the game and the optional information the player should know. Dialogue systems have mainly been designed to collaborate with the user to solve a particular task (Aust, H., Oerder, M., Seide, F. &Steinbiss, V., 1995), i.e. to handle dialogues that (Allen, et al., 2001) refer to as practical. For instance, game players can achieve certain goals or mission in the game, such as, orienting players to find a particular place in a 3D environment, or providing relevant information of missions at each level of the game through human-computer interaction. These systems usually operate in a specific and simple domain, which enables them to perform robustly. In our case study, we integrated this dialogue framework in a medical serious game. We are going to create an application for training doctors and medical students to interact with the virtual patient using natural language by the creation of a more realistic and immersive dialogue with virtual characters in games.

In 1950, mathematician Alan Turing proposed a question - "Can machines think?" (Turing, 2009). Since then, a number of researchers attempt to tackle this question. They attempt to visualize intelligence machine which now called chatbot or chatter robot. Chatbot is a brilliant technology that enables interactions between machine and man using human natural language. First chatbot introduced by Weizenbaum in 1966 (Weizenbaum, 1966), she was named ELIZA. She later became the main inspiration in computer science field for linguistic researchers. A huge breakthrough in chatbot technology then happened in 1995 when Dr. Richard Wallace, an ex-Professor of Carnegie Mellon University created an Artificial Linguistic Internet Computer Entity which named A.L.I.C.E (Wallace, 2009). A.L.I.C.E. later being described as a modern ELIZA. She was the three times winner for Loebner's annual instantiation of Turing's Test for machine intelligence (Shah, 2006).

When chatbot technology evolves, in an aspect of managing knowledge-based data (chatbot's brain), an evolvement in chatbots architecture can be justified. The first chatbot, ELIZA stored and embedded its data directly into application code, while more advanced A.L.I.C.E. uses custom design language, Artificial Intelligence Markup Language or AIML (a derivative of Extensible Markup Language or XML) to manage its datas (Shawar and Atwell, 2007)(Wallace, 2009). With the rapid evolution of Relational Database Model design together with Database Management System

(DBMS) technology, released more advance chatbots. One of the examples is VPbot, a SQL-Based chatbot designed for medical implementation (Ohno-Machado and Weber, 2005). VPbot was developed by Dr.Weber from Harvard University. It is a chatbot that takes advantage on Relational Database Model to store, manage and even use SQL language (database scripting language) to perform keyword and pattern matching process. Latest achievement in chatbot development, a further study has been done by Abbas Saliimawho's master in computer science. He created a chatbot for Diabetic patients in 2009. He then improved and named it as ViDi in "An Architecture Design of Virtual Dietitian (ViDi) for diabetic patients" in the following year. ViDichatbot was developed as an expansion towards E-CARE multimedia content.

Chatbot processing algorithms focus on the first developed chatbot (ELIZA), the most referred and used algorithms (A.L.I.C.E.'s AIML), the modern chatbot with the implementation of Relational Database Model design (VPbot) and the latest improved chatbot, ViDi.

ELIZA, the first chatbot developed by Professor Joseph Weizenbaum who's from Massachusetts Institute of Technology (MIT). It is described as a program that makes conversation with a computer by natural language possible. In ELIZA, input sentence are analyzed by the basis decomposition rules which are triggered by keywords within the input texts. Then, response are generated by reassembling rules associated from selected decomposition rules.

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) is an artificial intelligence natural language processing chat robot invented by Alan M.Turing in 1950. It can utilizes AIML, an XML language which designed for stimulus-response chat robots. Some view A.L.I.C.E. and AIML as an extension from the old ELIZA psychiatrist program. The comparison is fair regarding the stimulus-response architecture. But the A.L.I.C.E. bot has at present more than 40,000 categories of knowledge, whereas the original ELIZA had only about 200. Another innovation was provided by the web, which enabled natural language sample data collection possible on an unprecedented scale. A.L.I.C.E. won the Loebner Prize, an annual Turing Test, in 2000 and 2001. Although no computer has ever ranked higher than the humans in

the contest, she was ranked “most human computer” by the two panels of judges (Wallace, 2009).

VPbot, a SQL-Based chatbot designed for medical applications. VPbot stores ‘language rules’ into a relational data model. It's having many same features as A.L.I.C.E., but it is often easier to define new language rules in VPbot than that in AIML. Whereas A.L.I.C.E. is designed to produce generic responses to a wide range of topics.While VPbot is best suited to a targeted topic of conversation. The VPbot's algorithm accepts three types of input parameters, a vpid, the current topic, and a sentence. The vpid is a unique identifier for each VPbot instance. The output of VPbot is a new sentence or a new topic. Same with AIML, the output sentence can be dynamically constructed using parts of the input sentence; the database does not have to store every possible response (Ohno-Machado and Weber, 2005).

ViDi was developed as web-based application. Content presented in ViDi is originated from the collected diabetes educational data provided by medical professionals. The data stored in ViDi’s knowledge-based storage is temporary as It can be updated or edited by admin from time to time. Several additional approaches have been improved from ELIZA, A.L.I.C.E and VPbot in the development of ViDi, such as extension and prerequisite and OMAMC (One-Match and All-Match Categories) matching methods. The different is that ViDi’s All-match keywords had no limit over how many keywords can a single set have (VPbot limitation is three keywords for each single set).

OMAMC technique comprises of two components. 1) Keywords arrangement for matching precedence, and 2) keywords variety for matching flexibility. Describing the fundamental idea of OMAMC, One-match category describes a set of keyword which is build from a single word or a single phrase. While in All-match keywords matching category, keywords are presenting themselves individually which means that there will be many single words or many single phrases. Therefore, All-match keywords can be in a form of combination between single word and phrase, producing either multiple one-words keyword, multiple phrases keyword or both one-word/s and phrase/s keywords in a keywords set.

For both One-match and All-match categories, each keywords set will be stored as a single variable. Therefore, for All-match category which contains multiple

keywords within the same set, the symbol commas (“,”) is needed to separate each keywords. One-match is equal to an exact-match where it has to be exactly the same as the input sentence in terms of word/s and location. While All-match is equalized to a flexible-match where words’ location is a flexible factor. Which means the sequence of the keywords can be differ from the input sentence.

For the matching process, the precedence sequence will start from the keyword that had the most frequent word’s count to the keyword with lowest word’s count. Precedence word-match process is over OMAMC keyword search. While One-match keyword category will has the priority to be processed first then All-match category, as it holds the exact keywords data for matching including the sequence location of the words (exact match).

After comparing the four Chatbot model and their features from the previous studies, ViDichatbot will be selected as the reference model for this thesis and research development. While the techniques considered is OMAMC (One-Match and All-Match Categories) matching method.

3.0 METHODOLOGY

This chapter will cover the details and explanations on the methodology used to complete this project. In order to achieve the objective of the project and accomplish a perfect result, the methodology will be use is System Development Life Cycle (SDLC).

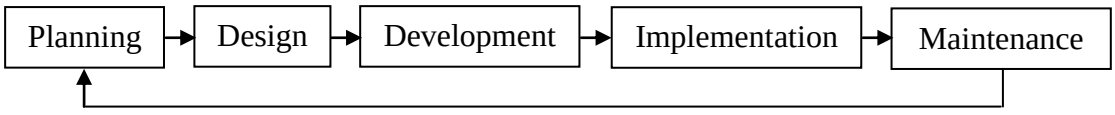

Figure 3.1 SLDC Phase

This project goes through three major steps during development process starting from planning, design and development. These phases had been identified to be the milestones in completing the subject. Each phase will undergo series of tasks that will contribute to the completion of the main task.

To identify all the information and requirements such as hardware and software, planning must be done in the proper manner. The planning phase has two main elements namely data collection and the requirements of hardware and software

In this phase, medical real-time education and management practice had been studied. The field study took place at Hospital Tengku Ampuan Afzan, Kuantan, Pahang, Malaysia. With the help from IIUM medical doctors and hospital staffs, several visits had been done to the patient/s-doctor session in which medical education and management with the real patients take place. Data collection is a stage in any area of study. At this stage I planned about the projects resources and requirements, literature studies and schedule to get more information in this study. All the materials are collected from journal, texts book and research papers gathered from libraries and Internet.

Chatbot is being coded using Hypertext Preprocessor (PHP) programming language together with Asynchronous Javascript + XML (AJAX) technology which contains the technology of Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), XMLHttp Request (XHR) and Document Object Model (DOM) that being accessed via JavaScript. jQueryjavascript's library also being used for DOM manipulation as well as CSS. As for database, Chatbot uses MySql database with phpMyAdmin as the Database Management System (DBMS). Chatbot had been tested across multiple web browsers including Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer. The hardware used for chatbot's development is Intel Core i5 750 Processor Window 7 Operating System.

4.0 DESIGN AND IMPLEMENTATION

This chapter will discuss on the overall approach and framework of the research. It describes method, technique or approach to be used in this project. It explains on the method used during design and implementation phases.

The following are the considered activities during the research development:

(i) Field study on medical and treatment education and previous study of prerequisite case study.

(ii) Design architectural model.

(iii) Develop prototype.

(iv) Testing.

4.1 Field Study On Medical And Treatment Education

Medical real-time education and management practice had been studied. The field study took place at Hospital Tengku Ampuan Afzan, Kuantan, Pahang, Malaysia. With the help from IIUM medical doctors and hospital staffs, several visits had been done to the patient/s-doctor session in which medical education and management with the real patients take place.

Figure 4.1 Patients-Doctor Diagnosis Session Field Study Live Shot

4.2 Previous Study On Literature Analysis Of Prerequisite Case Study

	ELIZA	A.L.I.C.E	VPbot	ViDi
Direct/exact relation/s between response/s	No	No	No	Yes
Matching precedence	Whole keyword's database with accordance to keyword's rank or precedence number	1. <that> tag 2. Others (whole keywords' database)	Whole keywords' database (lack of true support of context)	1. Extension response/s 2. Others (whole keywords' database)
Support for same keywords representing different meaning	Yes (same keywords can be used differently according to the Transformation Rules)	Yes (AIML complexity, writing previous utterance for each pattern is a tedious activity)	No	Yes (Extension and Prerequisite)
Possible link for the whole conversation	No (did not draw a path)	No (did not draw a path)	No (did not draw a path)	Yes (draw a path)
Human Working Memory imitation (storing previous utterance)	Not particularly (cannot store the whole utterance)	Yes (can store the whole utterance)	Not particularly (cannot store the whole utterance)	Yes (can store the whole utterance)
Longer keywords effect on matching precedence	No	No	Yes	Yes
Precedence analysis while keywords matching	While matching	While matching	After matching	While matching
Exact-match precedence priority against other matching types	N/A (matching are based on Decomposition Rules)	Highest (same category)	Highest (same category)	Highest (respective category)
Benchmark for stopping matching process if a match is found	Delimiter (comma/period) and/or higher RANK	No benchmark (stop instantly)	No benchmark (stop instantly)	Different category and/or lower word count
Generic keywords support	No	No	Yes (maximum three)	Yes (unlimited)
Response selection/generation precedence if there is a tie (more than one final response)	N/A (response generation is based on Transformation Rules)	1. Exact-match (atomic) 2. Other match (default and recursive)	1. Exact-match 2. Other match 3. Longest keywords	1. Exact-match 2. Other match 3. Longest Keywords 4. Lowest GWP value

Table 4.1 Chatbot Model Comparison

Chatbot processing algorithms focus on the first developed chatbot (ELIZA), the most referred and used algorithms (A.L.I.C.E.'s AIML), the modern chatbot with the implementation of Relational Database Model design (VPbot) and the latest improved chatbot, ViDi.

4.3 Data Collection And Content Development

To proceed this phase, me and my team had visit to Hospital Tengku Ampuan Afzan, Kuantan, Pahang to record some sample conversation of a doctor diagnose a patient. Their conversation has been used as a reference on how a verbal diagnosis is made and what question will normally asked and the possible answers from patient. From this, we can set up the database with potential questions and answers.

Patient: Good afternoon.
Doctor: Good afternoon. Have a seat. So, how are you today?
Patient: Thank you. I'm feeling ill, I've got quite a bad cough, but I don't seem to have a fever.
Doctor: I see. How long have you had these symptoms?
Patient: Oh, I've had the cough for two weeks, but feeling ill just these past few days.
Doctor: Are you having any other problems?
Patient: Well, I've got a headache. I've also had a little bit of diarrhea.
Doctor: Do you produce any phlegm when coughing?
Patient: Sometimes, but it's usually pretty dry.
Doctor: Do you smoke?
Patient: Yes, a few cigarettes a day. Certainly no more than a half a pack a day.
Doctor: How about allergies? Do you have any allergies?
Patient: Not that I'm aware of.
Doctor: Does your head feel stuffy?
Patient: Yes, for the past few days.
Doctor: You should stop smoking, you know?
Patient : OK, doctor. I'll try my best. Thanks.
Doctor : It's all right.

Patient : May I come in, Sir?

Doctor : Yes, you may.

Patient : I've been suffering from fever for the past two days.

Doctor : Did you take any medicine?

Patient : No, I didn't. I've got a cold too.

(Doctor examines the patient.)

Doctor : Your body temperature is 102° F. You must take medicines for at least three days.

Patient : Can I attend office?

Doctor : You shouldn't. You must take rest.

Patient : What about the diet?

Doctor : You should be on liquid diet.

Patient : Should I see you again?

Doctor : You needn't. But if the fever persists for more than three days, come and see me.

Patient : Sure.

Doctor : Here is the prescription.

Patient : Thank you, Sir.

Doctor : You're welcome.

Figure 4.2 Patient-Doctor Recorded Conversation

4.4 Architectural Model Design

Figure shows the schematic representation of chatbot architectural design model. In use case context, actor Users interact with chatbot by providing utterance input while it will react back by providing response output which correspond to that particular input. Actor Authors can add, edit or delete data from knowledge-based database contains of Root-Words, Synonyms, Keywords, and Responses tables data which have to be updated from time-to-time.

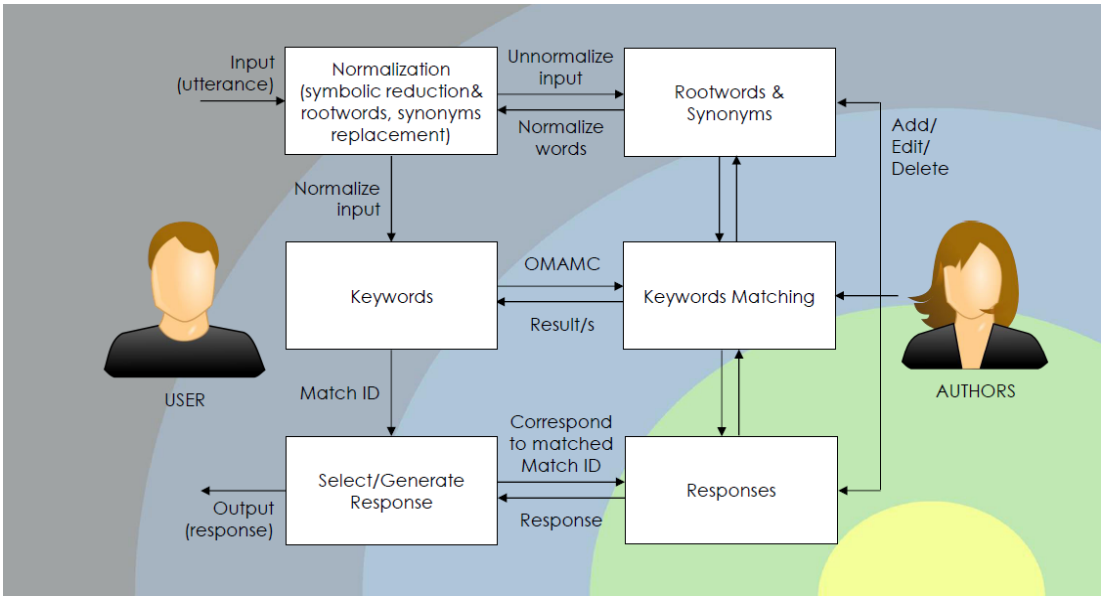

Figure 4.3 Chatbot Architecture Design Model

Generally, it takes about three main processes to generate an output from a user's input. The processes include 1) Normalization process, where input query is being normalized into standard form by removing unwanted symbols and words and synonyms replacements, 2) Matching process, is where input query are being matched with the keywords from the database, and lastly 3) Generate Response process, that correspond to the Responses database reflected according to the matched keywords. There are four tables are involved in this architecture. Rootwords and Synonyms tables to be uses in input normalize process, Keywords table to be used in keyword matching process, and Responses table that store every possible responses.

For the flow of whole process, it started with an user's input, then chatbot will perform Normalization onto the input query while interference with the Rootwords and Synonym table. Normalized words will be sent to Keywords Matching for matching process with the Keywords database table. This process involve OMAMC (One-Match and All-Match Categories) technique. All results have their own Match ID, then it will be sent to Select/Generate Response process where Responses database will execute the response which is associated with the Match ID back to the process as the final output. While in author context, author can perform add, edit, or delete action onto the database system.

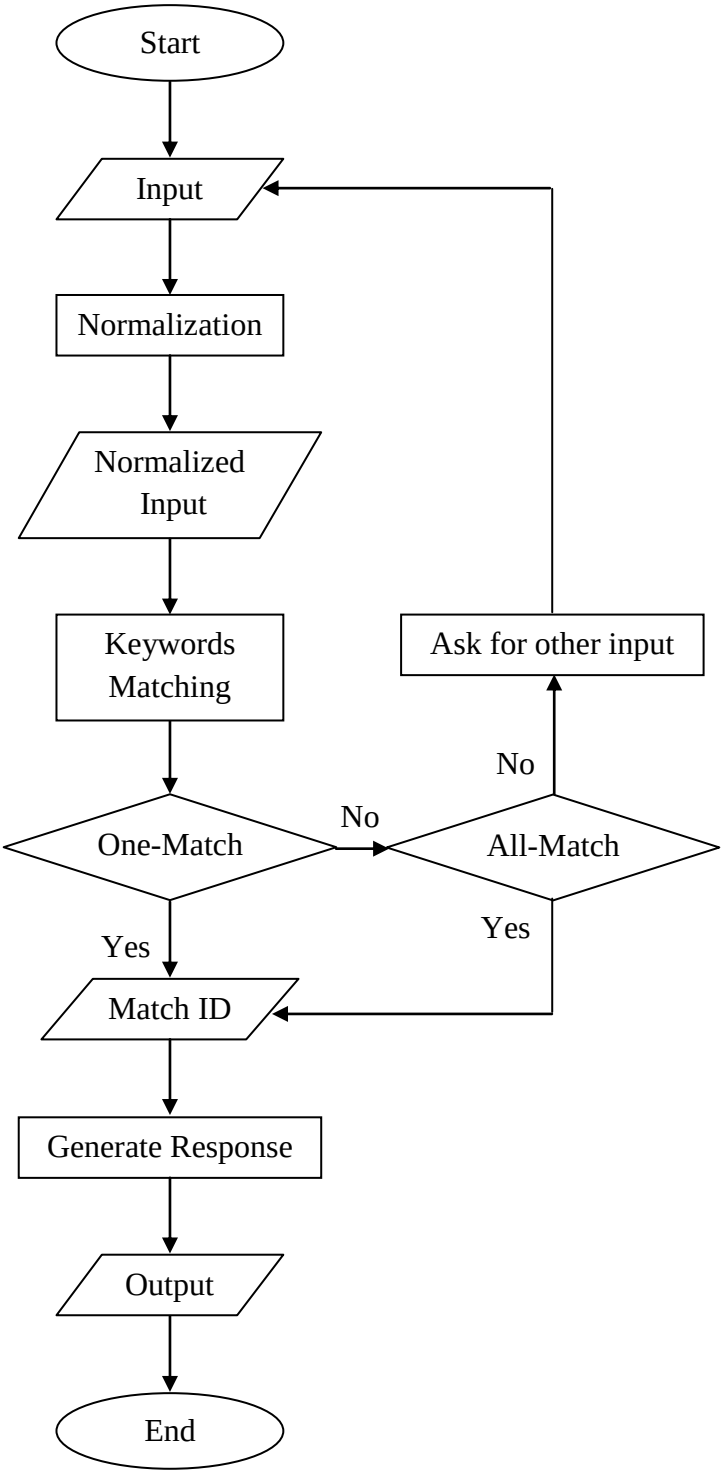

Figure 4.4 OMAMC (One-Match and All-Match Categories) Flow Chart

For OMAMC technique, when the process received the normalized input, it will refer to the One-Match database and search for the exact keywords, if not it will search from All-Match database and get the MatchID of the particular keywords. By holding the ID, it will then generate the correspond respond.

4.5 Development Of Prototype

CSS style sheet language is used to create chatbot interface.

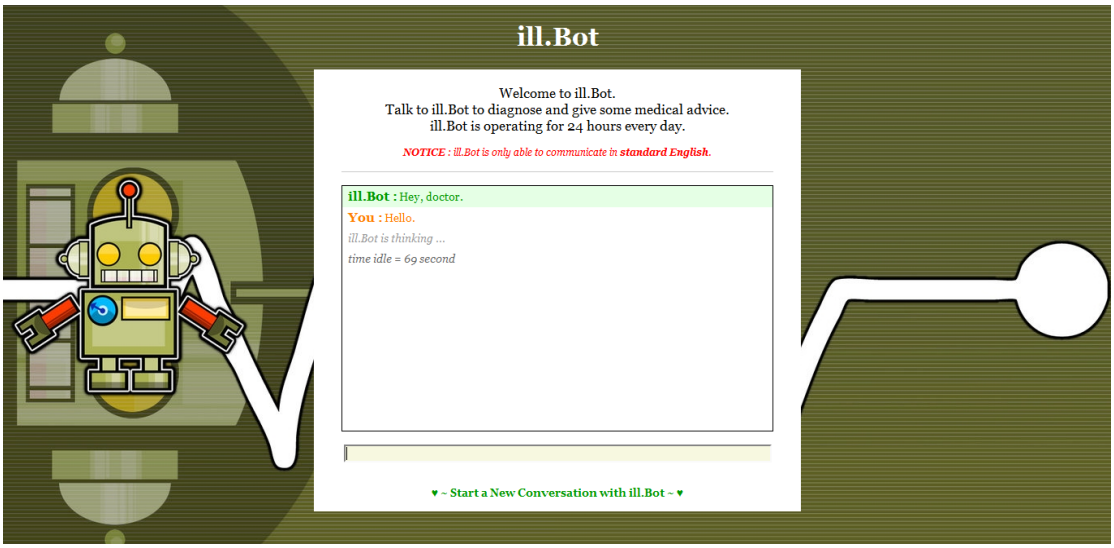

Figure 4.5 Chatbot Prototype Interface Design

Total of four tables in database.

1) keywords - table contains keywords for triggering responses.

2) responses - table contains potential reaction, reply or answer.

3) synonyms - having synonyms words to be swapped.

4) rootwords - used in user's input queries normalization process.

rootwords

Name	Data type	Description
word	text	Affixed or root/base word in input queries. Affixed word to be swap with “swap” while root/base word will be stayed with deleted affixation from original word in input queries.
swap	text	Root/base word to be swapped with “word” that is an affixed word in input queries.

synonyms

Name	Data type	Description
input	text	Word/phrase in input queries to be swapped.
swap	text	Word/phrase that will be swapped with “input” data in input queries.
wordCount	integer	Synonyms’ total word’s count (to determine the length of input words to be swap with synonyms data).

keywords

Name	Data type	Description
category	text	Keyword’s category (matched with response’s category).
words	text	Data that represents the keywords set. One-match or allmatch keywords’ types.
matchId	text	ID that matches variable “matchId” in table “responses”.
wordCount	integer	Keywords’ total word’s count (to determine the length of keywords).
allMatch	enum (‘yes’,‘no’)	‘yes’ for all-match or ‘no’ for one-match keywords’ types.
prerequisite	enum (‘yes’,‘no’)	‘yes’ for keywords with prerequisite data (matched with response’s prerequisite data).

responses

Name	Data type	Description
category	text	Response’s category. e.g. “diagnose”, “idle”, etc.
words	text	Data that represents the response.
matchId	text	ID that matches variable “matchId” in table “keywords”.
type	enum (‘A’,‘Q’,’S’)	Type of response, ‘A’ for answer, ‘Q’ for question or ‘S’ for statement.
extension	integer	Extension data for response (if any, one or many). Only response type ‘Q’ is eligible to have extension/s.
prerequisite	integer	Prerequisite data for response (if any, one or many).
general	float	General Words Percentage (GWP) value.

Table 4.2 Database Table Structure